#2

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

CERTIFIED COPY OF PRIORITY DOCUMENT

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

registration under the Companies Act does not constitute a new legal entity but merely ects the company to certain additional company law rules.

Signed J E Evans.

Dated 28 June 2001

Patents Form 7/77

The Patent Office

7/77

Patents Act 1977
Rule 15

Statement of inventorship and of right to grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
South Wales NP10 8QQ

1.	Your reference	GB920010010GB1
2.	Patent application number *(if you know it)*	**0108560.4**
3.	Full name of the or of each applicant	INTERNATIONAL BUSINESS MACHINES CORPORATION
4.	Title of invention	METHOD AND APPARATUS FOR ENCRYPTION OF DATA
5.	State how the applicant(s) derived the right from the inventor(s) to be granted a patent	By employment and agreement
6.	How many, if any, additional Patents Forms 7/77 are attached to this form?	
7.		I/We believe that the person(s) named over the page (and on any extra copies of this form) is/are the inventor(s) of the invention which the above patent application relates to. R D Moss Signature 4 April 2001 Date R D MOSS
8.	Name and daytime telephone number of person to contact in the United Kingdom	Anita Sekar Tel: 01962 818169

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

LAMBERT, Howard S.
UK resident
c/o IBM United Kingdom Limited
Intellectual Property Law
Hursley Park
Winchester
Hampshire SO21 2JN
England

6562102002

Patents ADP number *(if known)*

Patents ADP number *(if known)*

If there are more than three inventors, please write their names and addresses on the back of another Patents Form 7/77 and attach it to this form

Patents ADP number *(if known)*

REMINDER

Have you signed the form?

Patents Form 1/77

THE PATENT OFFICE
G
05 APR 2001
NEWPORT

The Patent Office

05APR01 E619802-1 D00611 1/77
P01/7700 0.00-0108560.4

Patents Act 1977
Rule 16

Request for grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
South Wales NP10 8QQ

1.	Your reference	GB920010010GB1
2.	Patent application number *(The Patent Office will fill in this part)*	0108560.4
3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	INTERNATIONAL BUSINESS MACHINES CORPORATION Armonk New York 10504 United States of America
	Patents ADP number *(if you know it)*	05196370001
	If the applicant is a corporate body, give the country/state of its incorporation	State of New York United States of America
4.	Title of the invention	METHOD AND APPARATUS FOR ENCRYPTION OF DATA
5.	Name of your agent *(if you have one)*	R D Moss
	"Address for Service" in the United Kingdom to which all correspondance should be sent *(including the postcode)*	IBM United Kingdom Limited Intellectual Property Department Hursley Park Winchester Hampshire S021 2JN
	Patents ADP number *(if you know it)*	06847966002

6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number	Country	Priority App No *(if you know it)*	Date of filing *(day/month/year)*

7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date or the earlier application	No of earlier application	Date of filing *(day/month/year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.)

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	16
Claim(s)	3
Abstract	1
Drawing(s)	6+6

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	2
Request for preliminary examination and search *(Patents Form 9/77)*	
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	

11. I/We request the grant of a patent on the basis of this application

R. D. Moss

Signature

4 April 2001
Date

R D MOSS

12. Name and daytime telephone number of person to contact in the United Kingdom

Anita Sekar
01962 818169

METHOD AND APPARATUS FOR ENCRYPTION OF DATA

This invention relates to a method and apparatus for encryption of data. In particular, the invention relates to encryption of streams of data using encryption based on Chaos Equations.

Data which is transmitted by any means, including along telecommunication links, via media such as storage disks, etc., must be protected to prevent the data being picked up and used by parties other than the intended recipient.

Security of information is a highly important aspect for any party transferring data. Users of networks, especially users of the Internet, are particularly vulnerable to unwanted parties intercepting data. The users may be commercial organisations, governments, universities or private individuals. Networks pass a huge variety of valuable, important and often confidential information. If the information is not secure, the consequences to the user can be disastrous. For example, the results may include financial losses, disclosure of confidential information, loss of confidence from clients and disruption to the user's activities.

In addition to transfer of information via networks, data transferred via other media such as storage disk, is also vulnerable to interception by unwanted parties.

To prevent such intrusion, data encryption methods are used to protect information during transmission from one end point to another. Encryption scrambles the data to make it unintelligible during transmission. In encryption systems, plain data is converted to a secure coded data (ciphertext) using an encryption method or algorithm with a secret key. A secret key is a usually a string of characters known only to the sender and the recipient. The recipient at the intended destination can decrypt the data by using the previously agreed secret key and the reverse of the encryption algorithm.

Data, such as binary data, text data and other forms of data which does not need to be delivered at a given rate, is encrypted in known cryptography systems in blocks of data. The data is broken into blocks of data. The blocks can be formed of a plurality of bytes of data and may be of varying length. Each block is encrypted according to an encryption

algorithm on a block by block basis. The decryption of the data is then carried out in a similar block by block manner.

When dealing with streams of data where the data is time dependent as opposed to blocks of text or binary data, block encryption is no longer appropriate or indeed possible. Streams of data include multi-media streams of voice, video, sensor data, and other types of data. This technology is applicable to pervasive computing, media streams, Internet music and video, command and control situations etc.

Streams may have a real time or data rate dependency, or may be sporadic and intermittent. Streams deliver data a byte at a time and may even have bytes skipped. Therefore, the bytes cannot be collected into blocks for encryption before delivery to the intended destination, as this will destroy the delivery rate and flow of the stream of data. For example, a stream of data may be communicated from a control environment in the form of one byte per week, a problem arises if the bytes must be collected into blocks before being communicated.

Current solutions buffer the data and encode the data using block ciphers. This can cause problems with real time or sparse streams.

An encryption system is needed that can operate on a byte per byte basis. This forces a type of encryption that is basically a byte substitution cipher, for example, a Caesar cipher and a Vigenere cipher. A Caesar cipher is a simple substitution cipher which uses an algorithm which shifts each letter in a message a certain number of spaces. An approach to cracking this form of cipher is to use statistical data about language letter frequencies. For example, the English language can be analysed to give a table of the frequency of occurrence of each letter in a text of say 1000 letters. An enciphered text can then be analysed to determine the letter frequencies and the frequencies compared to the known English language letter frequencies.

A more complex form of substitution cipher is the Vigenere cipher which is a polyalphabetic cipher. This form of cipher attempts to suppress the normal frequency data by using more than one alphabet to encrypt the message which results in a one to many relationship between each letter and its substitutes. The Vigenere cipher uses a table with each letter of the alphabet defining a row and each letter of the alphabet defining a column. The cipher table is used together with a keyword to encipher the message.

The keyword is repeated as many times as necessary above the plaintext message. For each letter of the plaintext message one finds the intersection of the row given by the corresponding keyword letter and the column given by the plaintext letter. Decryption is carried out by performing the reverse procedure, but the recipient needs to know the keyword.

Although harder to crack than simple Caesar ciphers, the Vigenere cipher can still be cracked by frequency analysis by locating bigrams in the ciphertext to determine the length of the keyword.

Substitution ciphers therefore have the disadvantage that they tend to be possible to crack using some form of frequency analysis.

The present invention uses non-linear dynamics and Chaos theory to prevent the frequency analysis approach of cracking the encryption. By the very nature of chaotic systems, they do not repeat and hence they are not susceptible to pattern or frequency analysis.

Fractal equations are one form of Chaos Equation and it is known to create encryption methods in visual cryptography using fractal equations. In normal visual cryptography, an image is encrypted by performing an XOR operation on the image with a key. The key can be a small image with randomly selected pixel colours. The key is XORed with the pixels of the actual image until all the image is encrypted. The image can be decrypted by XORing the encrypted image with the same key. Fractal visual encryption uses the same method with fractal geometry in Fractal Iteration of Information (FITIN) (http://www.cs.rit.edu/~nrr8953/fractal.html).

This form of visual encryption using fractals is based on symmetry which leads to reversibility in the encryption procedure. Due to physical limitations which restrict the use of the fractal geometry in visual encryption, the visual encryption achieved through this method is so far linear that it is not a good encryption for serious data.

In the present invention equations from Chaos theory, including fractals, are used to generate the encoding bytes for encrypting and decrypting data. There is no need for the concept of a block size or that every byte must be processed. Although the present invention has advantages in encrypting streams of data with a data rate dependency, the

encryption method of the present invention can also be used with data blocks with no rate dependency.

The normal meaning of "chaos" is *a condition or place of great disorder or confusion*, which sounds similar to the meaning of randomness: *having no specific pattern*. However, chaos as it relates to Chaos mathematics is very different. Most academic institutions will refer to chaos by a different name such as *non-linear dynamics*.

A chaotic system is not a random system, for example; a roulette wheel is a chaotic system not a random system because:

- To find out how high a ball will be after bouncing straight up and down against the ground for a certain amount of time. Find the height that the ball is dropped, the strength of gravity, etc. and
- use these numbers in the relevant equations to get the answer.
- A ball on a roulette wheel is a similar system with the addition that the table spins. The laws of motion state that it has one and only one final destination, it cannot be random. To find where on the roulette table a ball would land; find the height the ball was dropped, the speed and dimensions of the roulette table. Use this data in the appropriate equations to get the answer.

Therefore:

- o A chaotic system can be used to encrypt data, and the same system used to decrypt the data.
- o A random system can be used to encrypt data, but cannot decrypt the data since it is not possible to determine the matching random state (if it was possible, then it would not be random).

In this document, the term Chaos Equations is taken to include all forms of non-linear equations that are used to describe chaotic behaviour. There is an infinite set of such equations and only selected examples can be illustrated in this disclosure. Examples of Chaos Equations include, Fractal equations including Julia sets, Strange Attractors such as the Lorenz attractor, the Rossler attractor, the Henon attractor, the Gumowski/Mira attractor, the Tinkerbell attractor, the Periodic attractor, etc.

According to a first aspect of the present invention there is provided a method of encryption of data, in which the data is made up of a

series of data items, the method including the following steps: selecting a chaotic equation; defining starting conditions of the variables of the chaotic equation in the form of an input key; and applying the chaotic equation to each data item.

Preferably, the data is a continuous stream of data items. The stream of data items may have a rate dependency.

Optimally, the method includes an iterate step of updating the chaotic equation and the input key for each iteration value. An updated chaotic equation may be applied to each subsequent data item.

The data item may be a byte, a word or a dword.

Preferably, the step of applying the chaotic equation to the data item includes applying a modular arithmetic operation to combine the real and imaginary parts of the result of the chaotic equation and the data item. The encrypted data item may be calculated as $v \equiv (v \text{ xor } |z_{n+1}|) \bmod v_{max}$, where z_{n+1} is the value of the chaotic equation and v_{max} is the maximum value of v.

The chaotic equation may be one of a group that can comprise: Fractal equations, Julia sets, Lorenz attractor, Rossler attractor, Hénon attractor, Gumowski/Mira attractor and Tinkerbell attractor.

The defined variables of the equation may be the key to the encryption and are required at the encrypting source and the decrypting receiver.

Preferably, the method includes skipping data items by applying the chaotic equation to the data item and discarding the result.

According to a second aspect of the present invention there is provided an apparatus for encryption of data, in which the data is made up of a series of data items, the apparatus including: means for defining a chaotic equation; means for defining starting conditions of the variables of the chaotic equation in the form of an input key; and means for applying the chaotic equation to each data item.

Preferably, the data is a continuous stream of data items. The stream of data items may have a rate dependency.

The apparatus may include a plurality of defined chaotic equations.

Optimally, the apparatus includes an iterate means of updating the chaotic equation and the input key for each iteration value. The means for applying the chaotic equation to the data item may apply an updated chaotic equation to each subsequent data item.

The data item may be a byte, a word or a dword.

The means for applying the chaotic equation to the data item may include applying a modular arithmetic operation to combine the real and imaginary parts of the result of the chaotic equation and the data item. The encrypted data item may be defined as $v \equiv (v \text{ xor } |z_{n+1}|) \bmod v_{max}$, where z_{n+1} is the value of the chaotic equation and v_{max} is the maximum value of v.

The chaotic equation may be one of a group that can comprise: Fractal equations, Julia sets, Lorenz attractor, Rossler attractor, Hénon attractor, Gumowski/Mira attractor and Tinkerbell attractor.

The defined variables of the equation may be the key to the encryption and are required at the encrypting source and the decrypting receiver.

Preferably, the apparatus includes means for skipping data items by applying the chaotic equation to the data item and discarding the result.

According to a third aspect of the present invention there is provided a computer program product stored on a computer readable storage medium, comprising computer readable program code means for performing encryption of data made up of a series of data items, including for performing the following steps: selecting a chaotic equation; defining starting conditions of the variables of the chaotic equation as an input key; and applying the chaotic equation to each data item.

A method for encryption of data items is provided by defining a cipher key based on variables in a chaotic equation. The method includes selecting a chaotic equation, defining starting conditions of the variables of the equation, and applying the equation to each data item. The real and imaginary parts of the result of the iteration of the chaotic equation are combined with the data item by an arithmetic operation, for example and XOR

operation. Data items in a continuous stream with a rate dependency can be encrypted and decrypted on an item by item basis.

Embodiments of the invention will now be described, by means of example only, with reference to the accompanying drawings in which:

Figure 1 is a flow diagram representing the encryption method of the present invention;

Figure 2 is a plot of the Lorenz attractor used in the embodiment of Example 2 of the present invention;

Figure 3 is a plot of the Rossler attractor used in the embodiment of Example 3 of the present invention;

Figure 4 is a plot of the Hénon attractor used in the embodiment of Example 4 of the present invention;

Figure 5 is a representation of the Gumowski/Mira attractor used in the embodiment of Example 5 of the present invention; and

Figure 6 is a representation of the Tinkerbell attractor used in the embodiment of Example 6 of the present invention.

An encryption method is provided for encrypting and decrypting streams of data on a byte by byte basis using a key which is defined by a Chaos Equation. The key is defined by the equation and the parameters used, for example the starting point. The encrypted data can only be decrypted by a receiver with details of the equation used, the starting variables in the equation and the constants used in the equation.

Various specific examples are now described using a selection of Chaos Equations.

Example 1

Fractal equations are a type of Chaos Equation. Fractal geometry describes objects in non-integer dimensions. Fractal equations describe geometric figures with a property of invariance under a change of scale known as "self-similarity".

Imagine a three dimensional surface derived from a Fractal equation and draw round the edge of the surface for a chosen height (Note: Fractal edges are of infinite length). The line represents the key used to encode the data. In other words the key is defined by the:

- Fractal surface chosen
- Line start point
- Line height chosen
- Line direction

An example Fractal equation is the following:

$$z_{n+1} = f(z_n) = z_n^{\,2} + c$$

The value z_{n+1} is used to encode the data; it is not predictable without knowing the complex number c, the iteration number, the starting z_0 value and the actual equation used.

Setting the starting values of z_0 and c is equivalent to setting the cipher key i.e. setting the real and imaginary parts of $z_0 = (z_{0_x} + iz_{0_y})$ and $c = (c_x + ic_y)$.

When $|z_0| > 1$ and $|c| > 1$ (the desired case), the calculation tends towards □ (known as the □ attractor) therefore the values would overflow on a computer, to prevent this from happening modular arithmetic is used. Therefore, the equation becomes:

$$z_{n+1} \equiv (z_n^{\,2} + c) \bmod z_{\max}$$

Where the complex number $z_{\max}$ is the maximum allowed value of the complex number z.

In order to apply this encrypt or decrypt algorithm to the data item v (Note: v could be a *byte*, *word* or *dword*), we combine the real and imaginary parts of z_{n+1} e.g.

$$v \equiv (v \text{ xor } |z_{n+1}|) \bmod v_{\max}$$

Where v_{max} is the maximum value of v.

Other Fractal equations could be used, for example:

$$z_{n+1} = f(z_n) = z_n^{\,2} + (1+\varepsilon)e^{\frac{2\pi}{20}} z_n$$

$$= (z_{n_x} + iz_{n_y})^2 + (1+\varepsilon)\left(\cos\frac{2\pi}{20} + i\sin\frac{2\pi}{20}\right)(z_{n_x} + iz_{n_y})$$

$$= z_{n_x}^{\,2} - z_{n_y}^{\,2} + 2iz_{n_x}z_{n_y} + (1+\varepsilon)\left(\cos\frac{2\pi}{20}(z_{n_x} + iz_{n_y}) + \sin\frac{2\pi}{20}(iz_{n_x} - z_{n_y})\right)$$

The same process would be applied with the above equations using $v \equiv (v \text{ xor } |z_{n+1}|) \bmod v_{max}$ to provide a variation of the fractal cipher.

Example 1 - Algorithm

For the purpose of key generation we can assume that *c.r* and *c.i* are constant. Modular arithmetic is used to stop the real and imaginary parts of z_n from becoming excessively large since we will be using the □"attractor rather than the 0 attractor.

Therefore, to encrypt the data the calculated real and imaginary parts are combined with the data byte, the resultant is an encrypted data byte, hence we get the following:

```
{ complex number type definition                                    }
type
  TComplex = record
    r       : Extended;              { real part                    }
    i       : Extended;              { imaginary part               }
  end;
{ arbitrary starting condition values; note: z value will change    }
const
  c    : TComplex  = (r: 3.2616;           { parsec in light years }
                      i: 9.64846E4 );      { Faraday constant      }
  zmod      : TComplex  = (r: 1000000;     { real modulus          }
                      i: 1000000 );        { imaginary modulus     }
{ Absolute value of a complex number                                }
function ComplexAbs( a : TComplex ) : Extended;
```

```
    begin
      result := Sqrt( a.r * a.r + a.i * a.i );
    end;
  { complex remainder                                              }
  function Remainder( a : TComplex; b : TComplex) : TComplex;
    begin
      result.r := a.r - ( Trunc( a.r / b.r ) * b.r );
      result.i := a.i - ( Trunc( a.i / b.i ) * b.i );
    end;
  { encode/decode a byte          z(n+1) = z(n)*z(n) + c  }
  function Cipher( data : byte ) : byte;
    var
      temp     : TComplex;
    begin
      temp.r := (z.r * z.r) + (z.i * z.i) + c.r;
      temp.i := 2.0 * z.i * z.r + c.i;
      z      := Remainder ( temp, zmod );
      result := data xor Floor( ComplexAbs( temp ) );
    end;
```

The values assigned to *z.r, z.i, c.r* and *c.i* become the equivalent of the cipher key that is required at the encrypting source and decrypting receiver.

To handle the Julia set or any other Fractal equation the Cipher function is replaced with the appropriate calculation.

Figure 1 shows a flow diagram of the described encryption method. A byte 100 to be encrypted is selected. The chosen Chaos Equation 110 being used in the algorithm is run using an input key 140 which defines the variables of the Chaos Equation. The calculated real and imaginary parts of the Chaos Equation 110 are combined with the data byte 120 by an arithmetic operation, in this case an XOR operation. The updated Chaos Equation is saved 130 and the input key 140 is modified by communicating the result of the updated Chaos Equation via the iterate loop 150 in Figure 1. The coded byte 160 is returned.

Other arithmetic operations can be used at step 120 other than the XOR operation. For example, the values can be added or subtracted or modular multiplication or division could be used.

In the decryption process the reverse method is carried out. In some data delivery systems, delivery of all the bytes of data is not guaranteed and the delivery process may loose bytes of data. If a data byte is skipped and not received, the loop of the decryption process must still be carried out in order to correctly update the input key. In other words skipped bytes are handled by calling the iterate method and discarding the result. If a loop is missed due to skipped bytes, the decryption will become out of sync with the Chaos Equation and the decryption will fail.

The synchronisation between the bytes and the updating of the Chaos Equation via the input key also ensures that the data has not been tampered with in transit. Any change to the sequence or the bytes will result in an incorrect decryption.

Example 2

Chaotic encryption can be based on the Lorenz attractor. The Lorenz attractor consists of the following equations:

$$x_{n+1} = a(y_n - x_n)$$
$$y_{n+1} = rx_n - y_n - x_n z_n$$
$$z_{n+1} = x_n y_n - bz_n$$

A plot of the Lorenz attractor is given in Figure 2 with a = 16, b = 4, r = 45, x = 8 and z = 14.

The attractor will continue weaving back and forth between the two wings. Lorenz proved that complex, dynamical systems show order, but they **never repeat**.

The Lorenz attractor is important because, like all "well behaved" chaotic systems, the accuracy of any predictions about its future behavior decays exponentially as the interval between the present and the time for which predictions are being made increases. Another important quality of chaotic systems that is readily visible in this attractor is the sensitivity to change. If you change a single initial value of x, y, or z

by the slightest amount, the difference in the results will grow rapidly as time moves on.

```
const
  a    : Integer  = 16;
  b    : Integer  =  4;
  r    : Integer  = 45;
{ encode/decode a byte                                          }
function Cipher( data : byte ) : byte;
  var
    xp       : Integer;
    yp       : Integer;
  begin
    xp := (y - x) * a;
    yp := (r * x) - y - (x * z);
    z  := (x * y) - (b * z);
    y  := yp;
    x  := xp;
    Result := data xor (x + y - z);
  end;
```

Example 3

Chaotic encryption can be based on the Rossler attractor. The Rossler attractor is a simple set consisting of the following differential equations:

$$x_{n+1} = -y_n - z_n$$
$$y_{n+1} = x_n + ay_n$$
$$z_{n+1} = b + z_n(x_n - c)$$

A plot of the Rossler attractor is given in Figure 3 with a = 0.2, b = 0.2, c = 2.2

When the differential equations are graphed in 3-D space, they demonstrate what is known as banding. At c=2, there are two bands, the function follows these two bands, alternating between the two of them. This is because the attractor for the system is has a period of two. As c

increases, the period continues to double, and so do the bands. As *c* approaches 6, the number of periods goes to infinity and the attractor becomes chaotic.

```
const
  a    : Extended = 0.2;
  b    : Extended = 0.2;
  c    : Extended = 2.2;
{ encode/decode a byte                                          }
function Cipher( data : byte ) : byte;
  var
    xp       : Extended;
  begin
    xp := - y - z;
    y  := x + a * y;
    z  := (b + z * (x - c));
    x  := xp;
    Result := data xor Floor( z );
  end;
```

Example 4

Chaotic encryption can be based on the Hénon attractor. The Hénon map is a prototypical 2-D invertible iterated map with chaotic solutions proposed by the French astronomer Michel Hénon as a simplified model of the Poincare map for the Lorenz model.

The attractor is a simple set consisting of the following differential equations:

$$x_{n+1} = y_n + 1 - ax_n^2$$
$$y_{n+1} = bx_n$$

A plot of the Hénon attractor is given in Figure 4 with a = 1.4, b = 0.3. Each point on the display shows where the orbit of the system's 3-Dimensional strange attractor passes through the x-y plane. The plot is thus a slice of the complex orbit at a particular position around it. This was used to illustrate the chaotic aspects of a star's orbit around and through the galaxy or cluster to which it belongs.

However, it is not quite that straightforward. The pattern of a slice through such a complex orbit changes form according to the total energy of the system. The Hénon strange attractor is an attractor to which all these real-world attractors are attracted. It is formed from those other attractors when normal space is bent and folded to form a particular phase space in which this 'attractor of attractors' looks simple.

```
const
  a    : Extended = 1.4;
  b    : Extended = 0.3;
{ encode/decode a byte                                    }
function Cipher( data : byte ) : byte;
  var
    xp       : Extended;
  begin
    xp := (y + 1) - (a * x * x);
    y  := b * x;
    x  := xp;
    Result := data xor Floor( x + y );
  end;
```

Example 5

Chaotic encryption can be based on the Gumowski/Mira attractor. A formula is named after the two Physicists (or mathematicians) that discovered it. Their names are Gumowski and Mira. They did experiments at the CERN research facility in Geneva Switzerland. They were trying to calculate (or simulate) the trajectories of elementary particles like protons that move at high speeds in an accelerator, a circular channel with the diameter of a tin can but several meters long. Gumowski and Mira used the formula to simulate the orbits of the particles.

$$f(x) = ax + \frac{(1-a)2x^2}{1+x^2}$$
$$x_{n+1} = by_n + f(x_n)$$
$$y_{n+1} = -x_n + f(x_{n+1})$$

In which *a* is a parameter to be chosen usually anywhere in the range of -1 to 1. Parameter *b* is a very sensitive constant and usually stays at a value that is very close to 1.000. If the constant *b* is slightly increased to a

value of 1.001, then the trajectory will usually expand (or spiral outward to infinity). If the constant *b* is slightly decreased to something like a value of "0.999", then the trajectory will contract (or spiral inward) towards the attractor points.

A representation of the Gumowski/Mira attractor is given in Figure 5 with x = 19.945948645, y = 4.749808544, a = -0.669105405, b = 1.00001.

```
const
  a    : Extended;
  b    : Extended;
{ Calculate next x                                              }
function Fx( xn : Extended ) : Extended;
  begin
    Result := a * xn + ((1 - a) * 2 * xn * xn) / (1 + xn * xn);
  end;
{ encode/decode a byte                                          }
function Cipher( data : byte ) : byte;
  var
    xp       : Extended;
  begin
    xp := b * y + fx( x  );
    y  :=   - x + fx( xp );
    x  := xp;
    Result := data xor Floor( x + y );
  end;
```

Example 6

Chaotic encryption can be based on the so-called Tinkerbell attractor. This chaotic attractor is illustrated in Figure 6 and has a basin of attraction and periodic orbits with period smaller then or equal to 8.

$$\underline{Lx} = x_{n+1} = x_n^2 - y_n^2 + ax_n + by_n$$
$$Ly$$

$$\underline{Ly} = y_{n+1} = 2x_ny_n + cx_n + dy_n$$
$$Lx$$

In Figure 6, a = -0.7, b = -0.6013, c = 2.0, d =0.4 and there is a stable orbit at a = 0.485.

```
function Tinker (   data   :   byte   )   :   byte;
var
      xp   :  Extended;
begin
      xp   : = x * x - y * y + a * x + b * y;
      y    : = 2 * x * y + c * x + d * y;
      x    : = xp;
      Result   : = data xor Floor (  y   -   x  );
end;
```

It will be appreciated by a person skilled in the art, that any chaotic equation could be used to encrypt a stream of data using the method as described herein. A plurality of Chaos Equations can be defined in an encryption system and new equations can be added indefinitely. This particularly suits object-based systems. An encryption can only be decrypted by a person with the same defined Chaos Equation and the details of the starting variables and constants used in the equation.

Real time situations in which the described encryption method is highly useful include the following. Command and control security situations, for example, communication with a remote aircraft. Media stream such as those recorded on DVDs which include split streams which can be encrypted separately.

Although the present invention has advantages in encrypting streams of data with a data rate dependency, the encryption method of the present invention can also be used with data blocks with no rate dependency.

The present invention is typically implemented as a computer program product, comprising a set of program instructions for controlling a computer of similar device. These instructions can be supplied preloaded into a system or recorded on a storage medium such as a CD-ROM, or made available for downloading over a network such as the Internet or a mobile telephone network.

Improvements and modifications can be made to the foregoing without departing from the scope of the present invention.

CLAIMS

1. A method of encryption of data, in which the data is made up of a series of data items, the method including the following steps:
selecting a chaotic equation;
defining starting conditions of the variables of the chaotic equation in the form of an input key (140); and
applying the chaotic equation to each data item (110, 120, 130).

2. A method of encryption as claimed in claim 1, wherein the data is a continuous stream of data items.

3. A method of encryption as claimed in claim 2, wherein the stream of data items has a rate dependency.

4. A method of encryption as claimed in any one of claims 1 to 3, wherein the method includes an iterate step (150) of updating the chaotic equation (130) and the input key (140) for each iteration value.

5. A method of encryption as claimed in claim 4, wherein an updated chaotic equation is applied to each subsequent data item.

6. A method of encryption as claimed in any one of the preceding claims, wherein the data item is a byte, a word or a dword.

7. A method of encryption as claimed in any one of the preceding claims, wherein the step of applying the chaotic equation to the data item includes applying a modular arithmetic operation (120) to combine the real and imaginary parts of the result of the chaotic equation and the data item.

8. A method of encryption as claimed in claim 7, wherein the encrypted data item is defined as $v \equiv (v \text{ xor } |z_{n+1}|) \bmod v_{max}$, where z_{n+1} is the value of the chaotic equation and v_{max} is the maximum value of v.

9. A method of encryption as claimed in any one of the preceding claims, wherein the chaotic equation is one of a group that can comprise: Fractal equations, Julia sets, Lorenz attractor, Rossler attractor, Hénon attractor, Gumowski/Mira attractor and Tinkerbell attractor.

10. A method of encryption as claimed in any one of the preceding claims, wherein the defined variables of the equation are the key to the encryption and are required at the encrypting source and the decrypting receiver.

11. A method of encryption as claimed in any one of the preceding claims, wherein the method includes skipping data items by applying the chaotic equation to the data item and discarding the result.

12. An apparatus for encryption of data, in which the data is made up of a series of data items, the apparatus including:

means for defining a chaotic equation;

means for defining starting conditions of the variables of the chaotic equation in the form of an input key (140); and

means for applying the chaotic equation to each data item (110, 120, 130).

13. An apparatus as claimed in claim 12, wherein the data is a continuous stream of data items.

14. An apparatus as claimed in claim 13, wherein the stream of data items has a rate dependency.

15. An apparatus as claimed in any one of claims 12 to 14, wherein the apparatus includes a plurality of defined chaotic equations.

16. An apparatus as claimed in any one of claims 12 to 15, wherein the apparatus includes an iterate means (150) of updating the chaotic equation (130) and the input key (140) for each iteration value.

17. An apparatus as claimed in claim 16, wherein the means for applying the chaotic equation to the data item applies an updated chaotic equation to each subsequent data item.

18. An apparatus as claimed in any one of claims 12 to 17, wherein the data item is a byte, a word or a dword.

19. An apparatus as claimed in any one of claims 12 to 18, wherein the means for applying the chaotic equation to the data item includes applying a modular arithmetic operation (120) to combine the real and imaginary parts of the result of the chaotic equation and the data item.

20. An apparatus as claimed in claim 19, wherein the encrypted data item is defined as $v \equiv (v \text{ xor } |z_{n+1}|) \bmod v_{max}$, where z_{n+1} is the value of the chaotic equation and v_{max} is the maximum value of v.

21. An apparatus as claimed in any one of claims 12 to 20, wherein the chaotic equation is one of a group that can comprise: Fractal equations, Julia sets, Lorenz attractor, Rossler attractor, Hénon attractor, Gumowski/Mira attractor and Tinkerbell attractor.

22. An apparatus as claimed in any one of claims 12 to 21, wherein the defined variables of the equation are the key to the encryption and are required at the encrypting source and the decrypting receiver.

23. An apparatus as claimed in any one of claims 12 to 22, wherein the apparatus includes means for skipping data items by applying the chaotic equation to the data item and discarding the result.

24. A computer program product stored on a computer readable storage medium, comprising computer readable program code means for performing encryption of data made up of a series of data items, including for performing the following steps:

selecting a chaotic equation;

defining starting conditions of the variables of the chaotic equation as an input key (140); and

applying the chaotic equation to each data item (110, 120, 130).

ABSTRACT

METHOD AND APPARATUS FOR ENCRYPTION OF DATA

A method for encryption of data items is provided by defining a cipher key based on variables in a chaotic equation. The method includes selecting a chaotic equation (110), defining starting conditions of the variables of the equation (140), and applying the equation to each data item (120). The real and imaginary parts of the result of the iteration of the chaotic equation are combined with the data item by an arithmetic operation, for example and XOR operation (120). Data items in a continuous stream with a rate dependency can be encrypted and decrypted on an item by item basis.

FIG. 1

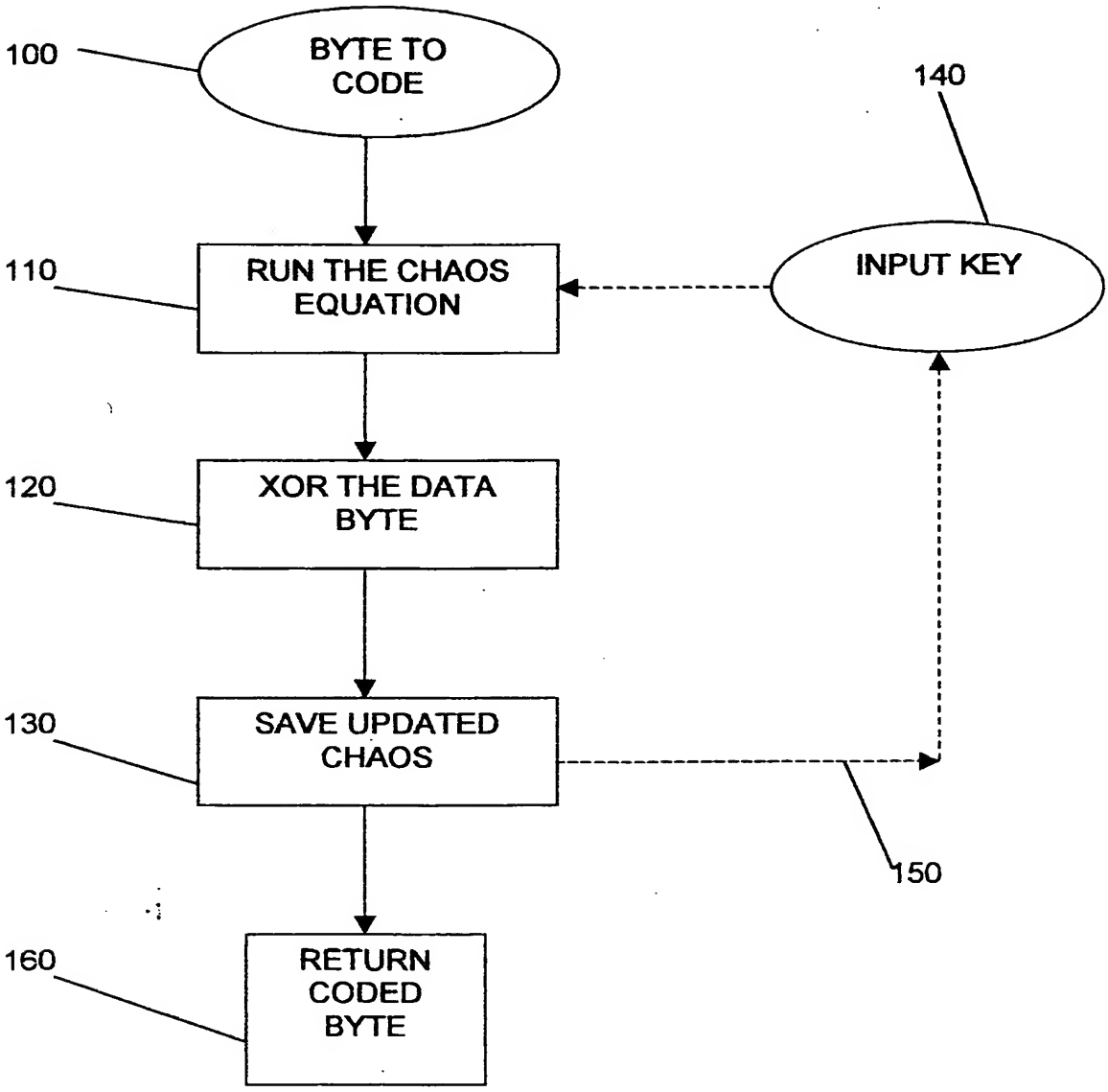
100
BYTE TO CODE
140
INPUT KEY
110
RUN THE CHAOS EQUATION
120
XOR THE DATA BYTE
130
SAVE UPDATED CHAOS
150
160
RETURN CODED BYTE

FIG. 2

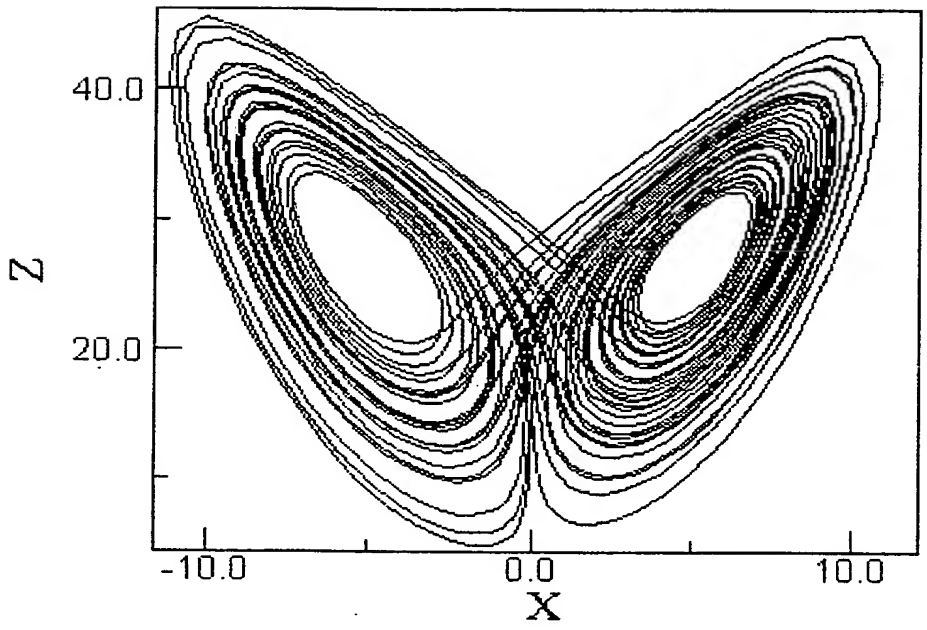

FIG. 3

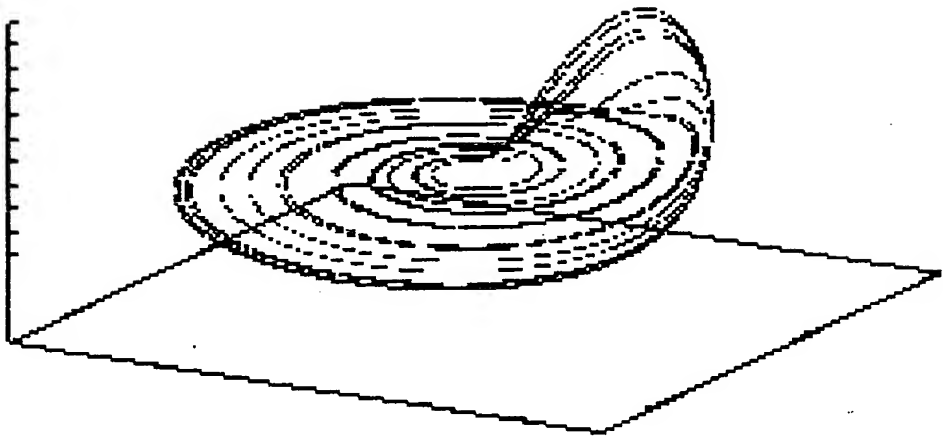

FIG. 4

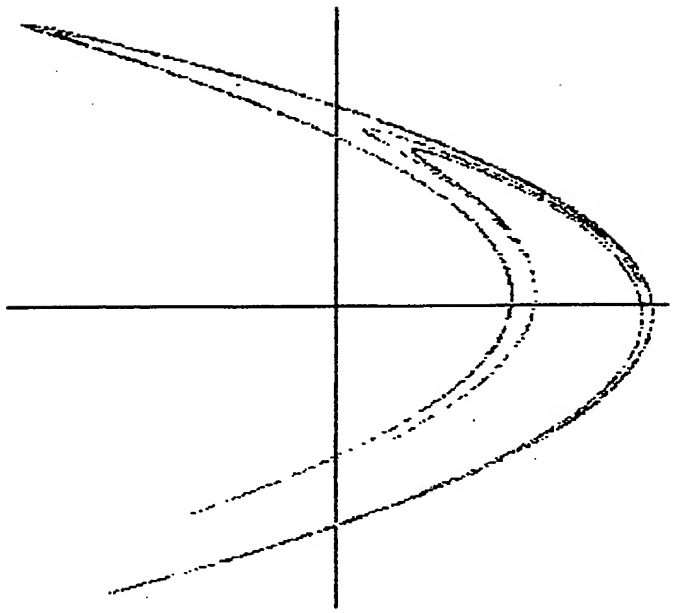

FIG. 5

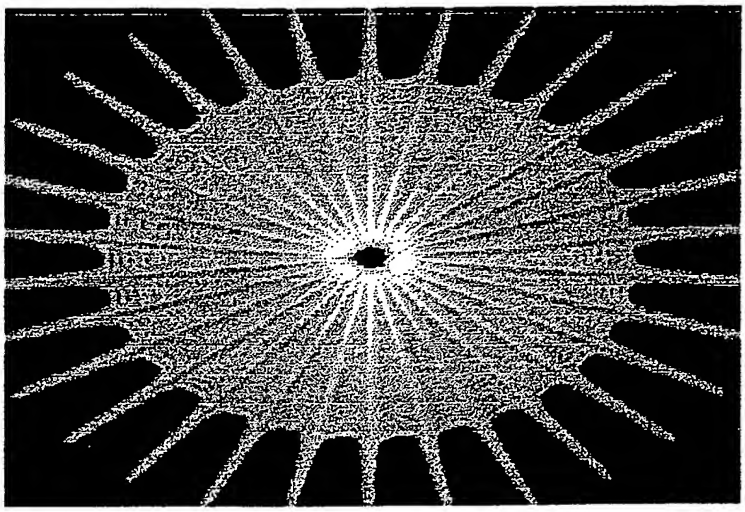

FIG. 6